

Construction of polar codes for arbitrary discrete memoryless channels

Talha Cihad Gulcu Min Ye Alexander Barg

Abstract

It is known that polar codes can be efficiently constructed for binary-input channels. At the same time, existing algorithms for general input alphabets are less practical because of high complexity. We address the construction problem for the general case, and analyze an algorithm that is based on successive reduction of the output alphabet size of the subchannels in each recursion step. For this procedure we estimate the approximation error as $O(\mu^{-1/(q-1)})$, where μ is the “quantization parameter,” i.e., the maximum size of the subchannel output alphabet allowed by the algorithm. The complexity of the code construction scales as $O(N\mu^4)$, where N is the length of the code.

We also show that if the polarizing operation relies on modulo- q addition, it is possible to merge subsets of output symbols without any loss in subchannel capacity. Performing this procedure before each approximation step results in a further speed-up of the code construction, and the resulting codes have smaller gap to capacity. We show that a similar acceleration can be attained for polar codes over finite field alphabets.

Experimentation shows that the suggested construction algorithms can be used to construct long polar codes for alphabets of size $q = 16$ and more with acceptable loss of the code rate for a variety of polarizing transforms.

Index terms: Channel degrading, Greedy symbol merging, Polarizing transforms.

I. INTRODUCTION

Arıkan’s polar codes [1] form the first explicit family of binary codes that achieve the capacity of binary-input channels. Polar codes rely on a remarkable phenomenon called channel polarization. After their introduction, both polar codes and the channel polarization concept have been used in a vast range of problems in information theory [2]–[11]. A drawback of the original proposal of [1] is that the construction of codes is not efficient because the alphabet of bit subchannels grows exponentially as a function of the number of iterations of the polarization procedure, resulting in an exponential complexity of construction.

The difficulty of selecting subchannels for information transmission with polar codes was recognized early on in a number of papers. According to an observation made in [12], the construction procedure of polar codes for binary-input channels relies on essentially the same density evolution procedure that plays a key role in the analysis of low-density parity-check codes. It was soon realized that the proposal of [12] requires increasing precision of the computations, but this paper paved way for later research on the construction problem.

An important step was taken in [13] which suggested to approximate each bit-channel after each evolution step by its degraded or upgraded version whose output alphabet size is constrained by a specified threshold μ that serves as a parameter of the procedure. As a result, [13] put forward an approximation procedure that results in a code not too far removed from the ideal choice of the bit-channels of [1]. This code construction scheme has a complexity of $O(N\mu^2 \log \mu)$, where $N = 2^n$ is the code length. For the channel degradation method described in [13], an error analysis and approximation guarantees are provided in [14].

Another approximation scheme for the construction of binary codes was considered in [15]. It is based on degrading each bit-channel after each evolution step, performed by merging several output symbols into one symbol based on quantizing the curve $p_{X|Y}(0|y)$ vs $h(p_{X|Y}(0|y))$, where $p_{X|Y}$ is the conditional distribution of the “reverse channel” that corresponds to the bit-channel in question. Symbols of the output alphabet that share the same range of quantization are merged into a single symbol of the approximating channel. Yet another algorithm based on bit-channel upgrading was described in [16], in which the authors argue that it is possible to obtain a channel which is arbitrarily close to the bit-channel of interest in terms of the capacity. However, no error or complexity analysis is provided in this work.

Moving to general input alphabets, let us mention a code construction algorithm based on degrading the subchannels in each evolution step designed in [17]. This algorithm involves a merging procedure of output symbols similarly to [15]. However, as noted by the authors, their construction scheme is practical only for small values of input alphabet size q ,

The authors are with Dept. of ECE and ISR, University of Maryland, College Park, MD 20742, USA. Emails: {tcgulcu,yeemmi}@gmail.com, abarg@umd.edu. A. Barg is also with Inst. Probl. Inform. Trans. (IITP), Moscow, Russia. Research supported in part by NSF grants CCF1217245 and CCF1422955.

its efficiency constrained by the complexity of order $O(\mu^q)$. Paper [18] proposed to perform the upgrading instead of degrading of the subchannels, but did not manage to reduce the implementation complexity. In [19], the authors consider another channel upgrading method for nonbinary-input channels, but stop short of providing an explicit construction scheme or error analysis.

Papers [20], [21], [22] addressed the construction problem of polar codes for AWGN channels. These works are based on Gaussian approximation of the intermediate likelihood ratios and do not analyze the error guarantees or rate loss of the obtained codes. A comparative study of various polar code constructions for AWGN channel is presented in [23]. Some other heuristic constructions for binary-input channels similar to the cited results for the Gaussian channel appear in [24], [25], [26]. Note also constructions of polar codes for some particular channels [27], [28], for various transformation kernels [29], [30], [31], and concatenated codes [32], [33].

In this paper we present a construction method of polar codes for input alphabets of arbitrary size, together with explicit analysis of approximation error and construction complexity. In particular, the complexity estimate of our procedure grows as $O(N\mu^4)$ as opposed to $O(N\mu^q)$ in earlier works. Our algorithm can be viewed as a generalization of the channel degradation method in [13] to nonbinary input channels. Although the approach and the proof methods here are rather different from earlier works, the estimate of the approximation error that we derive generalizes the error bound given by [14] for the binary case. Another interesting connection with the literature concerns a very recent result of [34] which derives a lower bound on the alphabet size μ that is necessary to restrict the capacity loss by at most a given value ϵ . This bound is valid for any approximation procedure that is based only on the degrading of the subchannels in each evolution step. The construction scheme presented here relies on the value μ that is not too far from this theoretical limit (see Proposition 3 for more details). We stress that we aim at approximating symmetric capacity of the channels, and do not attempt to construct or implement polar codes that attain Shannon capacity, which is greater than the symmetric one for non-symmetric channels.

Our paper is organized as follows. In Section II we give a brief overview of polar codes including various polarizing transformations for nonbinary alphabets. The rate loss estimate in the code construction based on merging pairs of output symbols in a greedy way is derived in Section III. In Section IV we argue that output symbols whose posterior probability vectors are cyclic shifts of each other can be merged with no rate loss. This observation enables us to formulate an improved version of the construction algorithm that further reduces the construction complexity. We have also implemented our algorithms and constructed polar codes for various nonbinary alphabets. These results are presented in Section V. For relatively small q we can construct rather long polar codes (for instance, going to length 10^6 for $q = 5$ takes several hours). For larger q such as 16 we can reach lengths of tens of thousands within reasonable time and with low rate loss. Even in this case, by increasing the gap to capacity of the resulting codes, we can reach lengths in the range of hundreds of thousands to a million without putting an effort in optimizing our software.

II. PRELIMINARIES ON POLAR CODING

We begin with a brief overview of binary polar codes. Let W be a channel with the output alphabet \mathcal{Y} , input alphabet $\mathcal{X} = \{0, 1\}$, and the conditional probability distribution $W_{Y|X}(\cdot|\cdot)$. Throughout the paper we denote the capacity and the symmetric capacity of W by $C(W)$ and $I(W)$, respectively. We say W is symmetric if $W_{Y|X}(y|1), y \in \mathcal{Y}$ can be obtained from $W_{Y|X}(y|0), y \in \mathcal{Y}$ through a permutation $\pi : \mathcal{Y} \rightarrow \mathcal{Y}$ such that $\pi^2 = \text{id}$. Note that if W is symmetric then $I(W) = C(W)$.

For $N = 2^n$ and $n \in \mathbb{N}$, define the polarizing matrix (or the Arıkan transform matrix) as $G_N = B_N F^{\otimes n}$, where $F = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$, \otimes is the Kronecker product of matrices, and B_N is a ‘‘bit reversal’’ permutation matrix [1]. In [1], Arıkan showed that given a symmetric and binary input channel W , an appropriate subset of the rows of G_N can be used as a generator matrix of a linear code that achieves the capacity of W as $N \rightarrow \infty$.

Given a binary-input channel W , define the channel W^N with input alphabet $\{0, 1\}^N$ and output alphabet \mathcal{Y}^N by the conditional distribution

$$W^N(y^N|x^N) = \prod_{i=1}^N W(y_i|x_i)$$

where $W(\cdot|\cdot)$ is the conditional distribution that defines W . Define a combined channel \widetilde{W} by the conditional distribution

$$\widetilde{W}(y^N|u^N) = W^N(y^N|u^N G_N).$$

In terms of \widetilde{W} , the channel seen by the i -th bit $U_i, i = 1, \dots, N$ (also known as the bit-channel of the i -th bit) can be written as

$$W_i(y^N, u_1^{i-1} | u_i) = \frac{1}{2^{n-1}} \sum_{\tilde{u} \in \{0,1\}^{n-i}} \widetilde{W}(y^N | (u_1^{i-1}, u_i, \tilde{u})). \quad (1)$$

We see that W_i is the conditional distribution of (Y^N, U_1^{i-1}) given U_i provided that the channel inputs X_i are uniformly distributed for all $i = 1, \dots, N$. Moreover, it is the case that [1] the bit-channels W_i can be constructed recursively using the channel transformations W^- and W^+ , which are defined by the equations

$$W^-(y_1, y_2 | u_1) \triangleq \frac{1}{2} \sum_{u_2 \in \{0,1\}} W(y_1 | u_1 + u_2) W(y_2 | u_2) \quad (2)$$

$$W^+(y_1, y_2, u_1 | u_2) \triangleq \frac{1}{2} W(y_1 | u_1 + u_2) W(y_2 | u_2). \quad (3)$$

The Bhattacharyya parameter $Z(W)$ of a binary-input channel W is defined as $Z(W) = \sum_{y \in \mathcal{Y}} \sqrt{W_{Y|X}(y|0)W_{Y|X}(y|1)}$. The bit-channels defined in (2)-(3) are partitioned into good channels $\mathcal{G}_N(W, \beta)$ and bad channels $\mathcal{B}_N(W, \beta)$ based on the values of $Z(W_i)$. More precisely, we have

$$\begin{aligned} \mathcal{G}_N(W, \beta) &= \{i \in [N] : Z(W_i) \leq \delta_N\} \\ \mathcal{B}_N(W, \beta) &= \{i \in [N] : Z(W_i) > 1 - \delta_N\}, \end{aligned} \quad (4)$$

where $[N] = \{1, 2, \dots, N\}$ and $\delta_N > 0$ is a small number. As shown in [35], for any binary-input channel W and any constant $\beta < 1/2$,

$$\begin{aligned} \lim_{N \rightarrow \infty} \frac{|\mathcal{G}_N(W, \beta)|}{N} &= I(W) \\ \lim_{N \rightarrow \infty} \frac{|\mathcal{B}_N(W, \beta)|}{N} &= 1 - I(W). \end{aligned} \quad (5)$$

Based on this equality, information can be transmitted over the good-bit channels while the remaining bits are fixed to some values known in advance to the receiver (in polar coding literature they are called *frozen bits*). The transmission scheme can be described as follows: A message of $k = |\mathcal{G}_N(W, \beta)|$ bits is written in the bits $u_i, i \in \mathcal{G}_N(W, \beta)$. The remaining $N - k$ bits are set to 0. This determines the sequence u^N which is transformed into $x^N = u^N G_N$, and the vector x^N is sent over the channel. Denote by y^N the sequence received on the output. The decoder finds an estimate of u^N by computing the values $\hat{u}_i, i = 1, \dots, N$ as follows:

$$\hat{u}_i = \begin{cases} \operatorname{argmax}_{u \in \{0,1\}} W_i(y^N, \hat{u}_1^{i-1} | u), & \text{if } i \in \mathcal{G}_N(W, \beta), \\ 0, & \text{if } i \in \mathcal{B}_N(W, \beta). \end{cases} \quad (6)$$

The results of [1], [35] imply the following upper bound on the error probability $P_e = \Pr(\hat{u}^N \neq u^N)$:

$$P_e \leq \sum_{i \in \mathcal{G}_N(W, \beta)} Z(W_i) \leq N 2^{-N^\beta} \quad (7)$$

where $\beta = \frac{1}{2} - \epsilon$, and $\epsilon > 0$ is arbitrarily small. This describes the basic construction of polar codes [1] which attains symmetric capacity $I(W)$ of the channel W with a low error rate. At the same time, (1), (6) highlight the main obstacle in the way of efficiently constructing polar codes: the size of the output alphabet of the channels W_i is of the order 2^{2N} , so it scales exponentially with the code length. For this reason, finding a practical code construction scheme of polar codes represents a nontrivial problem.

Concluding the introduction, let us mention that the code construction technique presented below can be applied to any polarizing transform based on combining pairs of subchannels. There has been a great deal of research on properties of polarizing operations in general. In particular, it was shown in [36] that (7) holds true whenever the input alphabet size q of the channel W is a prime number, and W^- and W^+ are defined as

$$W^-(y_1, y_2 | u_1) \triangleq \frac{1}{q} \sum_{u_2 \in \{0,1,\dots,q-1\}} W(y_1 | u_1 + u_2) W(y_2 | u_2) \quad (8)$$

$$W^+(y_1, y_2, u_1 | u_2) \triangleq \frac{1}{q} W(y_1 | u_1 + u_2) W(y_2 | u_2), \quad (9)$$

meaning that Arkan's transform $F = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ is polarizing for prime alphabets. For the case when q is a power of a prime, it was proved in [7] that there exist binary linear transforms different from F that support the estimate in (7) for some exponent β that depends on F . For example, [7] shows that the transform

$$G_\gamma = \begin{pmatrix} 1 & 0 \\ \gamma & 1 \end{pmatrix} \quad (10)$$

is polarizing whenever γ is a primitive element of the field \mathbb{F}_q . Paper [9] considered the use of Arkan's transform for the channels with input alphabet of size $q = 2^r$, showing that the symmetric capacities of the subchannels converge to one of $r + 1$ integer values in the set $\{0, 1, \dots, r\}$.

Even more generally, necessary and sufficient conditions for a binary operation $f : \mathcal{X}^2 \rightarrow \mathcal{X}^2$ given by

$$\begin{aligned} u_1 &= f(x_1, x_2), \\ u_2 &= x_2. \end{aligned} \quad (11)$$

to be a polarizing mapping were identified in [37], [38]. A simple set of *sufficient* conditions for the same was given in [39], which also gave a concrete example of a polarizing mapping for an alphabet of arbitrary size q . According to [39], in (11) one can take f in the form $f(x_1, x_2) = x_1 + \pi(x_2)$, where $\pi : \mathcal{X} \rightarrow \mathcal{X}$ is the following permutation:

$$\pi(x) = \begin{cases} \lfloor q/2 \rfloor, & \text{if } x = 0, \\ x - 1, & \text{if } 1 \leq x \leq \lfloor q/2 \rfloor, \\ x, & \text{otherwise.} \end{cases} \quad (12)$$

We include experimental results for code construction using the transforms (10) and (12) in Sect. V.

Finally recall that it is possible to attain polarization based on transforms that combine $l > 2$ subchannels. In particular, polarization results for transformation kernels of size $l \times l$ with $l > 2$ for binary-input channels were studied in [10]. Apart from that, [7] derived estimates of the error probability of polar codes for nonbinary channels based on transforms defined by generator matrices of Reed-Solomon codes. However, below we will restrict our attention to binary combining operations of the form discussed above.

III. CHANNEL DEGRADATION AND THE CODE CONSTRUCTION SCHEME

In the algorithm that we define, the subchannels are constructed recursively, and after each evolution step the resultant channel is replaced by its degraded version which has an output alphabet size less than a given threshold μ . In general terms, this procedure is described in more detail as follows.

Algorithm 1 Degrading of subchannels

input: DMC W , bound on the output size μ , code length $N = 2^n$, channel index i with binary representation

$i = \langle b_1, b_2, \dots, b_n \rangle_2$.

output: A DMC obtained from the subchannel W_i .

$T \leftarrow \text{degrade}(W, \mu)$

for $j = 1, 2, \dots, n$ **do**

if $b_j = 0$ **then**

$T \leftarrow T^-$

else

$T \leftarrow T^+$

end if

$T \leftarrow \text{degrade}(T, \mu)$

end for

return T

Before proceeding further we note that T^- and T^+ appearing in Algorithm 1 can be any transformations that produce combined channels for the polarization procedure. The possibilities range from Arkan's transform to the schemes discussed in the end of Section II.

The next step is to define the function `degrade` in such a way that it can be applied for general discrete channels. Ideally, the degrading-merge operation should optimize the degraded channel by attaining the smallest rate loss over all T' :

$$\inf_{\substack{T': T' \prec W \\ |\text{out}(T')| \leq \mu}} I(W) - I(T') \quad (13)$$

Equation (13) defines a convex maximization problem, which is difficult to solve with reasonable complexity. To reduce the computational load, [13] proposed the following approximation to (13): replace $y, y' \in \mathcal{Y}$ by a single symbol if the pair y, y' gives the minimum loss of capacity among all pairs of output symbols, and repeat this as many times as needed until the number of the remaining output symbols is equal to or less than μ (see Algorithm C in [13]). In [14], [15] this procedure was called *greedy mass merging*. In the binary case this procedure can be implemented with complexity $O(N\mu^2 \log \mu)$ because one can check only those pairs of symbols (y_1, y_2) which are closest to each other in terms of the likelihood ratios (see Theorem 8 in [13]). This simplification does not generalize to the channels with nonbinary inputs, meaning that we need to inspect all pairs of symbols. Since the total number of pairs is $O(\mu^4)$ after each evolution step, the overall complexity of the greedy mass merging algorithm for nonbinary input alphabets becomes $O(N\mu^4 \log \mu)$.

For a channel $W : \mathcal{X} \rightarrow \mathcal{Y}$ define

$$P_W(x|y) = \frac{W(y|x)}{\sum_{x_0 \in \mathcal{X}} W(y|x_0)},$$

$$P_Y(y) = \frac{1}{q} \sum_{x_0 \in \mathcal{X}} W(y|x_0)$$

for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. For a subset $A \subseteq \mathcal{Y}$, define

$$P_Y(A) = \sum_{y \in A} P_Y(y).$$

In the following lemma we establish an upper bound on the rate loss of the greedy mass merging algorithm for nonbinary input alphabets.

Lemma 1. *Let $W : \mathcal{X} \rightarrow \mathcal{Y}$ be a discrete memoryless channel and let $y_1, y_2 \in \mathcal{Y}$ be two output symbols. Let $\tilde{W} : \mathcal{X} \rightarrow \mathcal{Y} \setminus \{y_1, y_2\} \cup \{y_{\text{merge}}\}$ be the channel obtained from W by merging y_1 and y_2 which has the transition probabilities*

$$\tilde{W}(y|x) = \begin{cases} W(y|x), & \text{if } y \in \mathcal{Y} \setminus \{y_1, y_2\} \\ W(y_1|x) + W(y_2|x), & \text{if } y = y_{\text{merge}} \end{cases}.$$

Then

$$0 \leq I(W) - I(\tilde{W}) \leq \frac{P_Y(y_1) + P_Y(y_2)}{\ln 2} \sum_{x \in \mathcal{X}} |P_W(x|y_1) - P_W(x|y_2)|. \quad (14)$$

Proof: Since \tilde{W} is degraded with respect to W , we clearly have that $I(W) \geq I(\tilde{W})$, where $I(\cdot)$ is the symmetric capacity. To prove the upper bound for $I(W) - I(\tilde{W})$ in (14) let X be the random variable uniformly distributed on \mathcal{X} , and let Y be the random output of W . Then we have

$$\begin{aligned} I(W) - I(\tilde{W}) &= \left(H(X) - \sum_{y \in \mathcal{Y}} H(X|Y=y) P_Y(y) \right) \\ &\quad - \left(H(X) - H(X|Y \in \{y_1, y_2\}) (P_Y(y_1) + P_Y(y_2)) - \sum_{y \in \mathcal{Y} \setminus \{y_1, y_2\}} H(X|Y=y) P_Y(y) \right) \\ &= H(X|Y \in \{y_1, y_2\}) (P_Y(y_1) + P_Y(y_2)) \\ &\quad - H(X|Y=y_1) P_Y(y_1) - H(X|Y=y_2) P_Y(y_2). \end{aligned} \quad (15)$$

Next we have

$$\Pr(X=x|Y \in \{y_1, y_2\}) = \frac{\frac{1}{|\mathcal{X}|} (W(y_1|x) + W(y_2|x))}{P_Y(y_1) + P_Y(y_2)}$$

$$\begin{aligned}
&= \frac{\frac{1}{|\mathcal{X}|} W(y_1|x)}{P_Y(y_1) + P_Y(y_2)} + \frac{\frac{1}{|\mathcal{X}|} W(y_2|x)}{P_Y(y_1) + P_Y(y_2)} \\
&= \frac{P_Y(y_1)}{P_Y(y_1) + P_Y(y_2)} P_W(x|y_1) + \frac{P_Y(y_2)}{P_Y(y_1) + P_Y(y_2)} P_W(x|y_2) \\
&= \alpha_{12} P_W(x|y_1) + (1 - \alpha_{12}) P_W(x|y_2)
\end{aligned}$$

where $\alpha_{12} \triangleq \frac{P_Y(y_1)}{P_Y(y_1) + P_Y(y_2)}$. Hence, it follows from (15) that

$$\begin{aligned}
I(W) - I(\tilde{W}) &= (P_Y(y_1) + P_Y(y_2)) \sum_{x \in \mathcal{X}} [\alpha_{12} P_W(x|y_1) + (1 - \alpha_{12}) P_W(x|y_2)] \\
&\quad \times \log_2 \frac{1}{\alpha_{12} P_W(x|y_1) + (1 - \alpha_{12}) P_W(x|y_2)} \\
&\quad - P_Y(y_1) \sum_{x \in \mathcal{X}} P_W(x|y_1) \log_2 \frac{1}{P_W(x|y_1)} - P_Y(y_2) \sum_{x \in \mathcal{X}} P_W(x|y_2) \log_2 \frac{1}{P_W(x|y_2)}.
\end{aligned}$$

Rearranging the terms, we obtain

$$\begin{aligned}
I(W) - I(\tilde{W}) &= P_Y(y_1) \sum_{x \in \mathcal{X}} P_W(x|y_1) \log_2 \frac{P_W(x|y_1)}{\alpha_{12} P_W(x|y_1) + (1 - \alpha_{12}) P_W(x|y_2)} \\
&\quad + P_Y(y_2) \sum_{x \in \mathcal{X}} P_W(x|y_2) \log_2 \frac{P_W(x|y_2)}{\alpha_{12} P_W(x|y_1) + (1 - \alpha_{12}) P_W(x|y_2)}.
\end{aligned}$$

Next use the inequality $\ln x \leq x - 1$ to write

$$\begin{aligned}
I(W) - I(\tilde{W}) &\leq P_Y(y_1) \sum_{x \in \mathcal{X}} \frac{P_W(x|y_1)}{\ln 2} \left(\frac{P_W(x|y_1)}{\alpha_{12} P_W(x|y_1) + (1 - \alpha_{12}) P_W(x|y_2)} - 1 \right) \\
&\quad + P_Y(y_2) \sum_{x \in \mathcal{X}} \frac{P_W(x|y_2)}{\ln 2} \left(\frac{P_W(x|y_2)}{\alpha_{12} P_W(x|y_1) + (1 - \alpha_{12}) P_W(x|y_2)} - 1 \right)
\end{aligned}$$

which simplifies to

$$\begin{aligned}
I(W) - I(\tilde{W}) &\leq \frac{P_Y(y_1)}{\ln 2} \sum_{x \in \mathcal{X}} P_W(x|y_1) \frac{(1 - \alpha_{12})(P_W(x|y_1) - P_W(x|y_2))}{\alpha_{12} P_W(x|y_1) + (1 - \alpha_{12}) P_W(x|y_2)} \\
&\quad + \frac{P_Y(y_2)}{\ln 2} \sum_{x \in \mathcal{X}} P_W(x|y_2) \frac{\alpha_{12}(P_W(x|y_2) - P_W(x|y_1))}{\alpha_{12} P_W(x|y_1) + (1 - \alpha_{12}) P_W(x|y_2)}. \tag{16}
\end{aligned}$$

Bound the first term in (16) using the inequality

$$\left| \frac{(1 - \alpha_{12})(P_W(x|y_1) - P_W(x|y_2))}{\alpha_{12} P_W(x|y_1) + (1 - \alpha_{12}) P_W(x|y_2)} \right| \leq \frac{(1 - \alpha_{12}) |P_W(x|y_1) - P_W(x|y_2)|}{\alpha_{12} P_W(x|y_1)}$$

and do the same for the second term. We obtain the estimate

$$\begin{aligned}
I(W) - I(\tilde{W}) &\leq \frac{P_Y(y_1)}{\ln 2} \frac{1 - \alpha_{12}}{\alpha_{12}} \sum_{x \in \mathcal{X}} |P_W(x|y_1) - P_W(x|y_2)| \\
&\quad + \frac{P_Y(y_2)}{\ln 2} \frac{\alpha_{12}}{1 - \alpha_{12}} \sum_{x \in \mathcal{X}} |P_W(x|y_1) - P_W(x|y_2)| \\
&= \frac{P_Y(y_1) + P_Y(y_2)}{\ln 2} \|P_W(\cdot|y_1) - P_W(\cdot|y_2)\|_1.
\end{aligned}$$

This completes the proof of (14). ■

The bound (14) brings in metric properties of the probability vectors. Leveraging them, we can use simple volume arguments to bound the rate loss due to approximation.

Lemma 2. *Let the input and output alphabet sizes of W be q and M , respectively. Then, there exists a pair of output symbols (y_1, y_2) such that*

$$P_Y(y_1) = O\left(\frac{1}{M}\right), \quad P_Y(y_2) = O\left(\frac{1}{M}\right), \quad (17)$$

$$\|P_W(\cdot|y_1) - P_W(\cdot|y_2)\|_1 = O\left(\left(\frac{1}{M}\right)^{\frac{1}{q-1}}\right) \quad (18)$$

which implies the estimate

$$0 \leq I(W) - I(\tilde{W}) = O\left(\left(\frac{1}{M}\right)^{\frac{q}{q-1}}\right) \quad (19)$$

Proof: Consider the subset of output symbols $A_M(\mathcal{Y}) = \{y : P_Y(y) \leq 2/M\}$. Noticing that $|(A_M(\mathcal{Y}))^c| \leq M/2$, we conclude that

$$|A_M(\mathcal{Y})| \geq \frac{M}{2}. \quad (20)$$

Keeping in mind the bound (14), let us estimate the maximum value of the quantity

$$\min_{y_1, y_2 \in A_M(\mathcal{Y})} \|P_W(\cdot|y_1) - P_W(\cdot|y_2)\|_1. \quad (21)$$

For each $y \in \mathcal{Y}$, the vector $P_W(\cdot|y)$ is an element of the probability simplex

$$S_q = \left\{ (s_1, \dots, s_q) \in \mathbb{R}^q \mid s_i \geq 0, \sum_{i=1}^q s_i = 1 \right\}.$$

Let $R > 0$ be a number less than the quantity in (21). Clearly, for any $y_1, y_2 \in A_M(\mathcal{Y})$ the q -dimensional ℓ_1 -balls of radius $R/2$ centered at $P_W(\cdot|y_i), i = 1, 2$ are disjoint, and therefore, so are their intersections with S_q . Let $\text{vol}(S_q)$ be the $(q-1)$ -dimensional volume of S_q . It is easily seen that $\text{vol}(S_q) = \sqrt{q}/(q-1)!$, but in this proof we will stay with crude bounds (a more precise calculation is performed in the remark below). Clearly for any $y \in \mathcal{Y}$

$$\text{vol}\{B_{R/2}(P_W(\cdot|y_i)) \cap S_q\} = O\left(\frac{R}{2}\right)^{q-1}$$

On account of (20) we obtain that

$$\frac{M}{2} O\left(\left(\frac{R}{2}\right)^{q-1}\right) \leq \text{vol}(S_q) \quad (22)$$

whence

$$R = O\left(\left(\frac{1}{M}\right)^{1/(q-1)}\right)$$

for all R less than the quantity in (21). Hence, we see that there exist two output symbols $y_1, y_2 \in A_M(\mathcal{Y})$ such that the conditions (17), (18) hold simultaneously. So if these symbols are merged in the algorithm discussed, the rate loss is bounded above as in (19). \blacksquare

This lemma leads to an important conclusion for the code construction: to degrade the subchannels we should merge the symbols y_1, y_2 with small $P_Y(y_i)$ and such that the reverse channel conditional PMFs $P_W(\cdot|y_i), i = 1, 2$ are ℓ_1 -close. Performing this step several times in succession, we obtain the operation called `degrade` in the description of Algorithm 1. The properties of this operation are stated in the following proposition.

Proposition 3. *Let W be a DMC with input of size q .*

(a) *There exists a function $\text{degrade}(W, \mu)$ such that its output channel T satisfies*

$$0 \leq I(W) - I(T) \leq O\left(\left(\frac{1}{\mu}\right)^{\frac{1}{q-1}}\right). \quad (23)$$

(b) *For a given block length, let $W_N^{(i)}$ be the i -th subchannel after n evolution steps of the polarization recursion. Let $T_N^{(i)}$ denote the its approximation returned by Algorithm 1. Then*

$$0 \leq \frac{1}{N} \sum_{0 \leq i \leq N} (I(W_N^{(i)}) - I(T_N^{(i)})) \leq n O\left(\left(\frac{1}{\mu}\right)^{\frac{1}{q-1}}\right). \quad (24)$$

Proof: Let M be the cardinality of the output alphabet of W . Performing $M - \mu$ merging steps of the output symbols in succession, we obtain a channel with an output alphabet of size μ . If the pairs of symbols to be merged are chosen based on Lemma 2, then (18) implies that

$$\begin{aligned} 0 \leq I(W) - I(T) &\leq C(q) \sum_{i=\mu+1}^M \left(\frac{1}{i}\right)^{\frac{q}{q-1}} \\ &\leq C(q) \int_{\mu}^M (x-1)^{-\left(\frac{q}{q-1}\right)} dx = O\left(\left(\frac{1}{\mu}\right)^{\frac{1}{q-1}}\right) \end{aligned}$$

where $C(q)$ is a constant which depends on the input alphabet size q but not on the number n of recursion steps. This proves (23), and (24) follows immediately. \blacksquare

Remark III.1. This result provides a generalization to the nonbinary case of a result in [14] which analyzed the a merging (degrading) algorithm of [13]. For the case of binary-input channels, Lemma 1 of [14] gave an estimate $O(1/\mu)$ of the approximation error. Substituting $q = 2$ in (23), we note that this result is a generalization of [14] to channels with arbitrary finite-size input.

Remark III.2. Upper bounds similar to (23) are derived in [17, Lemma 6] and [18, Lemma 8]. The output symbol merging policy in [17] makes it possible to have $I(W) - I(\tilde{W}) = O((1/\mu)^{1/q})$. On the other hand, the channel upgrading technique introduced in [18] gives the same bound as (23). Recall that the code construction schemes considered in those two works have complexity $O(\mu^q)$. It is interesting to observe that merging a pair of output symbols at each step as we do here is as good as the algorithms based on binning of output symbols which requires a higher complexity.

Remark III.3. A very recent result of [34] states that any construction procedure of polar codes construction based on degrading after each polarization step, that guarantees the rate loss bounded as $I(W) - I(T) \leq \epsilon$, necessarily has the output alphabet of size $\mu = \Omega((1/\epsilon)^{\frac{q-1}{2}})$. Proposition 3 implies that the alphabet size of the algorithm that we propose scales as the square of this bound, meaning that the proposed procedure is not too far from being optimal, namely for any channel, our degradation scheme satisfies $\mu \leq (1/\epsilon)^{q-1}$, and there exists a channel for which $\mu \geq (1/\sqrt{\epsilon})^{q-1}$ holds true even for the optimal degradation scheme.

A HEURISTIC CALCULATION RELATED TO (22): Some of the implicit constants in the calculation that leads to (18) in the proof of Lemma 2 can be removed using the following (heuristic) geometric argument. Let

$$S_q = \left\{ x \in \mathbb{R}^q, \sum_{i=1}^q s_i \leq 1, s_i \geq 0, i = 1, \dots, q \right\}$$

be the regular q -dimensional simplex whose “outer” face is S_q . Consider the intersection of the q -dimensional balls with S_q rather than S_q . The volume of this intersection is the smallest when the center of the ball is located at a vertex of S_q . Let $V_q^p(R)$ be the volume of the ℓ_p -ball of radius R in q dimensions. Computing a crude estimate for the number of simplices that share a common vertex, note that they all fit in the ℓ_2 sphere of radius $\sqrt{2}$, so their number is at most $V_q^2(\sqrt{2})/\text{vol}(S_q)$. Assuming that the volume of the ℓ_1 -ball around the vertex is shared equally between these simplices, we estimate the volume of the intersection to be

$$V_q^1(R/2) \frac{\text{vol}(S_q)}{V_q^2(\sqrt{2})} = \frac{(\Gamma(2)R)^q}{\Gamma(q+1)} \frac{\Gamma\left(\frac{q}{2}+1\right)}{(2\sqrt{2}\Gamma(3/2))^q} \text{vol}(S_q).$$

Using a packing argument similar to (22), we obtain

$$\frac{R^q}{q!} \frac{\Gamma\left(\frac{q}{2}+1\right)}{(2\sqrt{2}\Gamma(3/2))^q} \leq \frac{2}{M}$$

which gives

$$R \leq C \left(\frac{1}{M}\right)^{1/q}$$

where $C = 2\sqrt{2}\Gamma\left(\frac{3}{2}\right)\left(\frac{2q!}{\Gamma\left(\frac{q}{2}+1\right)}\right)^{1/q} \leq \sqrt{2\pi}q$. This calculation results in a bound slightly weaker than the one in (23), but contains no implicit constants.

IV. NO-LOSS ALPHABET REDUCTION

Throughout this section we will use the transformation (8)-(9), in which the “+” is addition modulo q . We discuss a way to further reduce the complexity of the code construction algorithm using the additive structure on \mathcal{X} . As shown in (14), the symmetric capacity loss is small if the posterior distributions induced by the merged symbols are ℓ_1 -close. Here we argue that if these vectors are related through cyclic shifts, the output symbols can be merged at no cost to code performance.

Consider the construction of q -ary polar codes for channels with input alphabet $q \geq 2$. Since $I(W) = \log q - H(X|Y)$, to construct polar codes it suffices to track the values of $H(X|Y)$ for the transformed channels. Keeping in mind that $H(X|Y) = E(-\log P_{X|Y}(X|Y))$, let us write the polarizing transformation in terms of the reverse channel $P_{X|Y}$:

$$\left. \begin{aligned} P_{Y-}^{-}(y_i, y_j) &= P_Y(y_i)P_Y(y_j), \\ P_{X|Y-}^{-}(x|y_i, y_j) &= \sum_{u_2 \in \mathcal{X}} P_{X|Y}(x + u_2|y_i)P_{X|Y}(u_2|y_j) \\ P_{Y+}^{+}(u, y_i, y_j) &= \left(\sum_{x \in \mathcal{X}} P_{X|Y}(u + x|y_i)P_{X|Y}(x|y_j) \right) P_Y(y_i)P_Y(y_j), \\ P_{X|Y+}^{+}(x|u, y_i, y_j) &= \frac{P_{X|Y}(u + x|y_i)P_{X|Y}(x|y_j)}{\sum_{x_0 \in \mathcal{X}} P_{X|Y}(u + x_0|y_i)P_{X|Y}(x_0|y_j)} \end{aligned} \right\} \quad (25)$$

If P_X is uniform, both P_X^{+} and P_X^{-} are also uniform. Consequently, the transformation (25) is the same as (8)-(9) under the uniform prior distributions. Throughout this section we will calculate the transformation of probability distributions using (25) instead of (8)-(9) since we rely on the posterior distributions to merge symbols.

Definition IV.1. Given a distribution P_{XY} on $\mathcal{X} \times \mathcal{Y}$, define an *equivalence relation* on \mathcal{Y} as follows: $y_1 \stackrel{P}{\sim} y_2$ if for every $x \in \mathcal{X}$ there exists $x_1 \in \mathcal{X}$ such that $P_{X|Y}(x + x_1|y_1) = P_{X|Y}(x|y_2)$. This defines a partition of \mathcal{Y} into a set of equivalence classes $\mathcal{Y} = \{A_1, A_2, \dots, A_{|\mathcal{Y}|}\}$.

We show that if $y_1 \stackrel{P}{\sim} y_2$, then we can merge y_1 and y_2 into one alphabet symbol without changing $H(X|Y)$ for all P_{XY}^s , $s \in \{-, +\}^n$ and all $n \geq 1$. As a consequence, it is possible to assign one symbol to each equivalence class, i.e., the effective output alphabet of W for the purposes of code construction is formed by the set \mathcal{Y} .

To formalize this intuition, we need the following definitions.

Definition IV.2. Consider a pair of distributions P_{XY_1}, Q_{XY_2} . We say that two subsets of output alphabets $A \subseteq \mathcal{Y}_1, B \subseteq \mathcal{Y}_2$ are in correspondence, denoted $A \simeq B$, if

- (1) $P_{Y_1}(A) = P_{Y_2}(B)$;
- (2) For every $y_1 \in A$ and $y_2 \in B$ and every $x \in \mathcal{X}$ there exists $x_1 \in \mathcal{X}$ such that $P_{X|Y_1}(x + x_1|y_1) = Q_{X|Y_2}(x|y_2)$ (the value of x_1 may depend on y_1 and y_2).

Note that condition (2) in this definition implies that all the elements in A are in the same equivalence class, and all the elements in B are also in the same equivalence class.

Definition IV.3. We call the distributions P_{XY_1}, Q_{XY_2} *equivalent*, denoted $P_{XY_1} \equiv Q_{XY_2}$, if there is a bijection $\phi : \mathcal{Y}_1 \rightarrow \mathcal{Y}_2$ such that $A \simeq \phi(A)$ for every equivalence class $A \in \mathcal{Y}_1$.

Note that two equivalent distributions have the same $H(X|Y)$.

The following proposition underlies the proposed speedup of the polar code construction. Its proof is computational in nature and is given in the Appendix.

Proposition 4. Let P_{XY_1}, Q_{XY_2} be two distributions. If $P_{XY_1} \equiv Q_{XY_2}$ then for all $s \in \{-, +\}^n, n \geq 1$ we have $P_{XY_1}^s \equiv Q_{XY_2}^s$ (and therefore $H_{P^s}(X|Y_1) = H_{Q^s}(X|Y_2)$).

The next proposition provides a systematic way to merge output symbols of the synthesized channels obtained by the ‘+’ transformation.

Proposition 5. Let distribution P_{XY} on $\mathcal{X} \times \mathcal{Y}$, and let P_{XY-}^{-} and P_{XY+}^{+} be defined as in (25). For every $(v, y_1, y_2) \in \mathcal{X} \times \mathcal{Y}^2$ we have

$$(v, y_1, y_2) \stackrel{P^{+}}{\sim} (-v, y_2, y_1), \quad (26)$$

where if $y_1 = y_2$ then $v \neq 0$.

Proof: For every $y_1, y_2 \in \mathcal{Y}$ and any $u_1, u \in \mathcal{X}$, we have

$$\begin{aligned} P_{X|Y^+}^+(u|(u_1, y_1, y_2)) &= \frac{P_{X|Y}(u_1 + u|y_1)P_{X|Y}(u|y_2)}{\sum_{x_0 \in \mathcal{X}} P_{X|Y}(u_1 + x_0|y_1)P_{X|Y}(x_0|y_2)} \\ &= \frac{P(-u_1 + (u + u_1)|y_2)P(u_1 + u|y_1)}{\sum_{x_0 \in \mathcal{X}} P_{X|Y}(-u_1 + (u_1 + x_0)|y_2)P_{X|Y}(u_1 + x_0|y_1)} \\ &= P_{X|Y^+}^+(u + u_1|(-u_1, y_2, y_1)). \end{aligned}$$

This proves (26). ■

No-loss cyclic merging algorithm

Using the above considerations, we can reduce the time needed to construct a polar code. The informal description of the algorithm is as follows. Given a DMC $W : \mathcal{X} \rightarrow \mathcal{Y}$, we calculate a joint distribution P_{XY} on $\mathcal{X} \times \mathcal{Y}$ by assuming a uniform prior on \mathcal{X} . We then use (25) to recursively calculate $P_{XY^s}^s$, and after each step of the recursion we reduce the output alphabet size by assigning one symbol to the whole equivalence class. Namely, for each equivalence class A in the output alphabet \mathcal{Y}^s , we set $P_{Y^s}^s(A) = \sum_{y \in A} P_{Y^s}^s(y)$ and $P_{X|Y^s}^s(x|A) = P_{X|Y^s}^s(x|y^*)$ for an arbitrarily chosen $y^* \in A$. Note that y^* can be chosen arbitrarily because the vectors $P_{X|Y^s}^s(\cdot|y), y \in A$ are cyclic shifts of each other. By Prop. 4, we have $I(W^s) = \log q - H_{P^s}(X|Y)$, i.e., the alphabet reduction entails no approximation of the capacity values.

Let us give an example, which shows that this simple proposal can result in a significant reduction of the size of the output alphabet. Let W be a q -ary symmetric channel (qSC) $W : \mathcal{X} \rightarrow \mathcal{Y}, |\mathcal{X}| = |\mathcal{Y}| = q$

$$W(y|x) = (1 - \epsilon)\delta_{x,y} + \frac{\epsilon}{q-1}(1 - \delta_{x,y}), \quad (27)$$

and let us take $q = 4$. Consider the channels $W^s, s \in \{+, -\}^n$ obtained by several applications of the recursion (2)-(3). The actual output alphabet size of the channels W^+, W^{++} and W^{+++} is $4^3, 4^7$, and 4^{15} , respectively. At the same time, the effective output alphabet size of W^+, W^{++} and W^{+++} obtained upon merging the equivalence classes in \mathcal{Y} is no more than 3, 24, and 1200. In particular, the effective output alphabet size of W^{+++} is less than a 10^6 -th fraction of its actual output alphabet size. Let $n \geq 3$ and $s \in \{+, -\}^n$. If s starts with $+++$, then the effective output alphabet size of W^s is less than a $(10^{6 \times 2^{n-3}})$ -th fraction of its actual alphabet size.

Improved greedy mass merging algorithm

Now we are ready to describe the improved code construction scheme. Prop. 4 implies that if the vectors $P_{X|Y}(\cdot|y_i), i = 1, 2$ are cyclic shifts of each other, merging them into one symbol \tilde{y} incurs no rate loss. Extending this intuition, we assume that performing greedy mass merging using all the cyclic shifts of these vectors improves the accuracy of the approximation.

Given a DMC $W : \mathcal{X} \rightarrow \mathcal{Y}$, we calculate a joint distribution P_{XY} on $\mathcal{X} \times \mathcal{Y}$ by assuming the uniform prior on \mathcal{X} and taking W as the conditional probability. We then use (25) to recursively calculate $P_{XY^s}^s$ and after each step of transformation:

- (1) If the last step in s is $+$: First use the `merge_pair` function below to merge the symbols (u_1, y_1, y_2) and $(-u_1, y_2, y_1)$ for all u_1, y_1, y_2 , then use the `degrade` function below on $P_{XY^s}^s$.
- (2) If the last step in s is $-$, use the `degrade` function below on $P_{XY^s}^s$.

The function `merge_pair`($Q, (y_1, y_2, u)$) is defined as follows: Form the alphabet $\tilde{\mathcal{Y}} = \mathcal{Y} \setminus \{y_1, y_2\} \cup \{\tilde{y}\}$, putting $Q_{\tilde{Y}}(y) = Q_Y(y), Q_{X|\tilde{Y}}(x|\tilde{y}) = Q_{X|Y}(x|y)$ for all $x \in \mathcal{X}$ and $y \in \mathcal{Y} \setminus \{y_1, y_2\}$ and

$$\begin{aligned} Q_{\tilde{Y}}(\tilde{y}) &= Q_Y(y_1) + Q_Y(y_2), \\ Q_{X|\tilde{Y}}(x|\tilde{y}) &= \frac{Q_Y(y_1)Q_{X|Y}(x|y_1) + Q_Y(y_2)Q_{X|Y}(x + u|y_2)}{Q_{\tilde{Y}}(\tilde{y})}. \end{aligned} \quad (28)$$

Remark IV.1. Due to the concavity of the entropy function [40, Thm. 2.7.3], $H(X|Y)$ can only increase after calling the `merge_pair` function.

Algorithm 2 The degrade function

input: distribution P_{X,Y_0} over $\mathcal{X} \times \mathcal{Y}_0$, the target output alphabet size μ .

output: distribution $Q_{X,Y}$ over $\mathcal{X} \times \mathcal{Y}$, where $|\mathcal{Y}| \leq \mu$.

```

 $Q \leftarrow P$ 
 $\ell \leftarrow |\mathcal{Y}|$ 
while  $\ell > \mu$  do
   $(y_1, y_2, u) \leftarrow \text{choose}(Q)$ 
   $Q \leftarrow \text{merge\_pair}(Q, (y_1, y_2, u))$ 
   $\ell \leftarrow \ell - 1$ 
end while
return  $Q$ 

```

The function $\text{choose}(Q)$ is defined as follows. Find the triple y_1, y_2 and $u \in \mathcal{X}$ such that the change of conditional entropy $H_Q(X|Y)$ incurred by the merge $(y_1, y_2) \rightarrow \tilde{y}$ using $\text{merge_pair}(Q, (y_1, y_2, u))$

$$\Delta(H) \triangleq Q_{\tilde{Y}}(\tilde{y})H(X|\tilde{Y} = \tilde{y}) - \sum_{i=1}^2 Q_Y(y_i)H(X|Y = y_i)$$

is the smallest among all the triples $(y_i, y_j, u) \in \mathcal{Y}^2 \times \mathcal{X}$.

Remark IV.2. The main difference between Algorithm 2 and the ordinary greedy mass merging algorithm discussed in Sect. III (e.g., Algorithm C in [13]) can be described as follows. In order to select a pair of symbols that induces the smallest increase of $H(X|Y)$, Algorithm 2 considers all the cyclic shifts of the posterior distributions of pairs of symbols, while the “ordinary” greedy mass merging algorithm examines only the distributions themselves. As argued above, this is the reason that Algorithm 2 leads to a smaller rate loss than Algorithm 1.

Note that to perform the ‘+’ transformation, we first use (26) to merge pairs of symbols with cyclically shifted posterior vectors and then switch to greedy mass merging. In doing so, we incur a smaller rate loss because the number of steps of approximation performed for Algorithm 2 is only half the number of steps performed in Algorithm 1. Moreover, since (26) provides a systematic way of merging symbols with cyclically shifted distributions, (in other words, we do not need to search all the pairs in order to find them,) the running time of Algorithm 2 is also reduced from that of Algorithm 1. This intuition is confirmed in our experiments which show that the overall gap to capacity of the constructed codes is smaller than the one attained by using the basic greedy mass merging, while the time taken by the algorithm is reduced from greedy mass merging alone. More details about the experiments are given in Sect. V.

The finite field transformation of [7]

We also note that Prop. 4 remains to be valid when the alphabet is a finite field \mathbb{F}_q and Arıkan’s transform $F = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ is replaced by the transform given by (10). This fact is stated in the proposition below. Its proof is similar to Prop. 4 and will be omitted.

Proposition 6. *Let $\mathcal{X} = \mathbb{F}_q$ and let P_{XY_1}, Q_{XY_2} be two distributions. If $P_{XY_1} \equiv Q_{XY_2}$ then for all $s \in \{-, +\}^n, n \geq 1$ we have $P_{XY_1^s}^s \equiv Q_{XY_2^s}^s$ (and therefore $H_{P^s}(X|Y_1) = H_{Q^s}(X|Y_2)$).*

V. EXPERIMENTAL RESULTS

There are several options of implementing the alphabet reduction procedures discussed above. The overall idea is to perform cyclic merging (with no rate loss) and then greedy mass merging for every subchannel in every step $n \geq 1$ of the recursion.

Greedy mass merging (the function `degrade` of Algorithm 1) calls for finding a pair of symbols y_1, y_2 whose merging minimizes the rate loss $\tilde{\Delta}$, which can be done in time $O(M^2 \log M)$, $M := |\mathcal{Y}|$. In practice this may be too slow, so instead of optimizing we can merge the first pair of symbols for which the rate loss is below some chosen threshold C . It is also possible to merge pairs of symbols based on the proximity of probabilities on the RHS of (14).

Note also that greedy mass merging can be applied to any binary polarizing operation including those described in Sect. II. We performed a number of experiments using addition modulo q , the finite field polarization G_γ , and a polarizing operation from [39]. A selection of results appears in Fig. 1. In Examples 1-3 we construct polar codes for

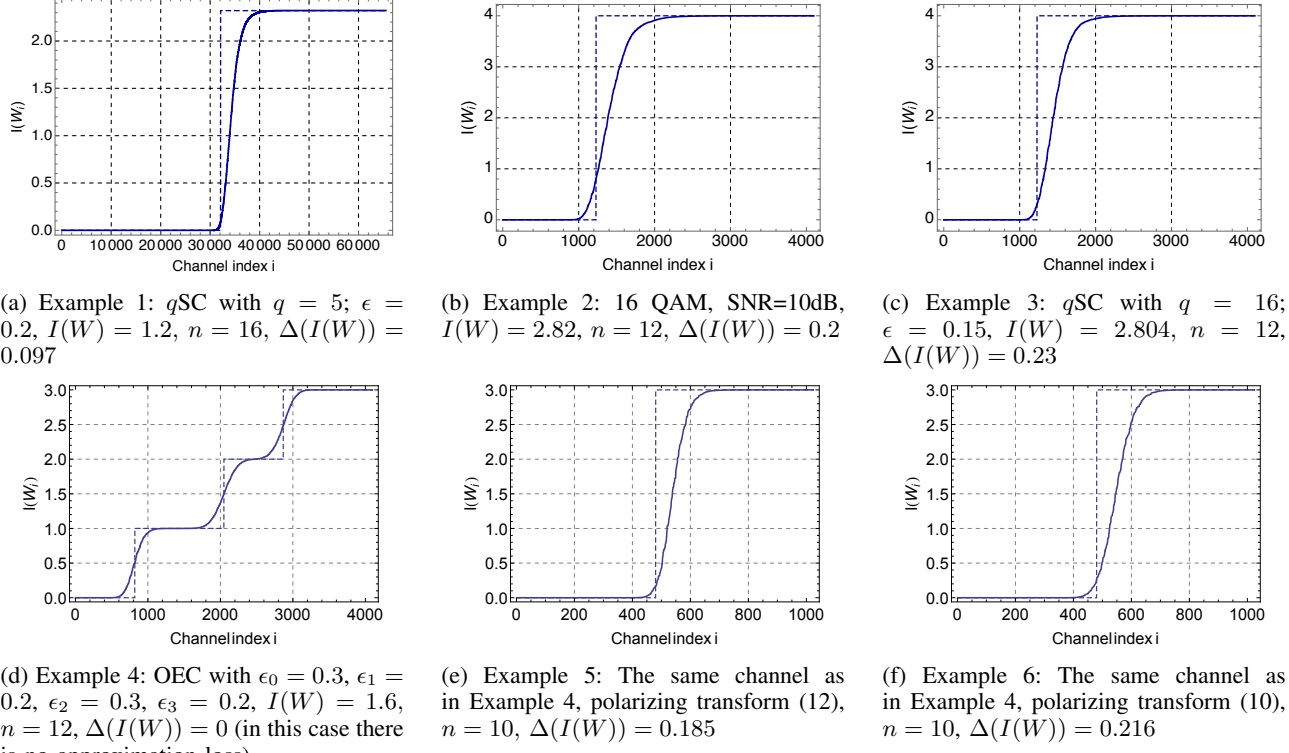


Fig. 1: Construction of nonbinary polar codes. In Fig. (a)-(c) we plot the capacity distribution of subchannels for channels with $q = 5$ and 16 (in these examples q SC is a q -ary symmetric channel defined in (27)). In Examples 4-6 we apply different polarizing transforms, showing convergence to different number of extremal configurations for the same channel (here OEC is the ordered erasure channel, see (29))

the q -ary symmetric channel (27) and the 16 QAM channel, showing the distribution of capacities of the subchannels. In Examples 4-6 we apply different polarizing transforms to a channel W with inputs $\mathcal{X} = \{0, 1\}^3$ and outputs $\mathcal{Y} = \{0, 1\}^3 \cup \{? * *, ? *, ???\}$, where $*$ can be 0 or 1. The transitions are given by

$$\begin{aligned} W(x_1 x_2 x_3 | x_1 x_2 x_3) &= 0.3, & W(? x_2 x_3 | x_1 x_2 x_3) &= 0.2 \\ W(?? x_3 | x_1 x_2 x_3) &= 0.3, & W(??? | x_1 x_2 x_3) &= 0.2 \end{aligned} \quad (29)$$

for all $x_1, x_2 \in \{0, 1\}$. Following [9], we call W an *ordered erasure channel*. One can observe that under the addition modulo- q transform (8)-(9) the channel polarizes to several extremal configurations, while under the transforms given in (10), (12) it converges to only two levels. This behavior, predicted by the general results cited in Section II, supports the claim that the basic algorithm of Sect. III does not depend on (is unaware of) the underlying polarizing transform. More details about the experiments are provided in the captions to Fig. 1.

It is interesting to observe that the q -ary symmetric channel for $q = 16$ polarizes to two levels under Arıkan's transform. In principle there could be 5 different extremal configurations, and it is a priori unclear that no intermediate levels arise in the limit. An attempt to prove this fact was previously made in [36], but no complete proof is known to this date.

Next we give some simulation results to support the conclusions drawn for Algorithm 2. We construct polar codes of several block lengths for q SC W with $q = 4$ and $\epsilon = 0.15$, setting the threshold $\mu = 256$. The capacity of the channel equals $I(W) = 1.15242$.

N	t_1	t_2	ΔI_1	ΔI_2	$\frac{t_1}{t_2}$	$\frac{\Delta I_1}{\Delta I_2}$
128	404	177	0.041	0.026	2.3	1.6
256	1038	490	0.048	0.033	2.1	1.5
512	2256	1088	0.055	0.038	2.1	1.5
1024	4378	2164	0.061	0.042	2.0	1.5

In this table N is the code length, t_1 is the running time of greedy mass merging and t_2 is the running time of Algorithm 2 (our algorithm) in seconds. The quantities ΔI_1 and ΔI_2 represent the rate loss (the gap between $I(W)$ and the average capacity of the subchannels) in greedy mass merging and our algorithm, respectively.

Binary codes: Here our results imply the following speedup of Algorithm A in [13]. Denote $LR(y) = W(y|1)/W(y|0)$. The cyclic merging means that we merge any two symbols $(y_1, y_2) \rightarrow \tilde{y}$ if $LR(y_1) = LR(y_2)^{\pm 1}$, so we can only record the symbols $y \in \tilde{Y}$ with $LR(y) \geq 1$. This implies that the threshold μ in [13] can be reduced to $\mu/2$. Overall the alphabet after the $+$ or $-$ step is reduced by a factor of about 8 while the code constructed is exactly the same as in [13]. In the following table we use the threshold values $\mu = 32$ for [13] and $\mu = 16$ for our algorithm. The codes are constructed for the BSC channel with $\epsilon = 0.11$.

N	t_A	t_2	$\frac{t_A}{t_2}$	N	t_A	t_2	$\frac{t_A}{t_2}$
512	3.6	0.5	7.2	1024	7.3	1.1	6.6
2048	14.7	2.3	6.4	4096	29.2	4.6	6.3

In the second table N is the code length, t_A is the running time of Algorithm A in [13], and t_2 is the running time of our algorithm in seconds. Our algorithm indeed is about 7 times faster, and the codes constructed in both cases are exactly the same.

VI. CONCLUSION

We considered the problem of constructing polar codes for nonbinary alphabets. Constructing polar codes has been a difficult open question since the introduction of the binary polar codes in [1]. Ideally, one would like to obtain an explicit description of the polar codes for a given block length, but this seems to be beyond reach at this point. As an alternative, one could attempt to construct the code by approximating each step of the recursion process. For binary codes, this has been done in [13],[14], but extending this line of work to the nonbinary case was an open problem despite several attempts in the literature. We take this question one step closer to the solution by designing an algorithm that approximates the construction for moderately-sized input alphabets such as $q = 16$. The algorithm we implement works for both binary and non-binary channels with complexity $O(N\mu^4)$, where N is the blocklength and μ is the parameter that limits the output alphabet size. Furthermore, the error estimate we derive generalizes the estimate of [14] to the case of nonbinary input alphabets (but relies on a different proof method). It is also interesting to note that the error is rather close to a *lower bound* for this type of construction algorithms, derived recently in [34]. Apart from presenting a theoretical advance, this algorithm provides a useful tool in the analysis of properties of various polarizing transforms applied to nonbinary codes over alphabets of different structure. The proposed construction algorithm also brings nonbinary codes closer to practical applications, which is another promising direction to be explored in the future.

APPENDIX: PROOF OF PROP. 4

We will show that if $P_{XY_1} \equiv Q_{XY_2}$, then $P_{XY_1^-}^- \equiv Q_{XY_2^-}^-$ and $P_{XY_1^+}^+ \equiv Q_{XY_2^+}^+$, which will imply the full claim by induction on n .

(a) (The ‘ $-$ ’ case) The distributions $P_{XY_1^-}^-$ and $Q_{XY_2^-}^-$ are defined on the sets $\mathcal{X} \times \mathcal{Y}_1^2$ and $\mathcal{X} \times \mathcal{Y}_2^2$, respectively. In order to prove that $P_{XY_1^-}^- \equiv Q_{XY_2^-}^-$, we need to show that for every $A_1, B_1 \in \mathcal{Y}_1$, we have $A_1 \times B_1 \simeq \phi(A_1) \times \phi(B_1)$. Indeed,

$$\begin{aligned}
 \sum_{(y_1, y_2) \in A_1 \times B_1} P_{Y_1^-}^-(y_1, y_2) &= \sum_{y_1 \in A_1} \sum_{y_2 \in B_1} P_{Y_1^-}^-(y_1, y_2) \\
 &= \sum_{y_1 \in A_1} \sum_{y_2 \in B_1} P_{Y_1}(y_1) P_{Y_1}(y_2) \\
 &= \left(\sum_{y_1 \in A_1} P_{Y_1}(y_1) \right) \left(\sum_{y_2 \in B_1} P_{Y_1}(y_2) \right).
 \end{aligned}$$

Similarly,

$$\sum_{(y_1, y_2) \in \phi(A_1) \times \phi(B_1)} Q_{Y_2^-}^-(y_1, y_2) = \left(\sum_{y_1 \in \phi(A_1)} Q_{Y_2}(y_1) \right) \left(\sum_{y_2 \in \phi(B_1)} Q_{Y_2}(y_2) \right).$$

Since $A_1 \simeq \phi(A_1)$ and $B_1 \simeq \phi(B_1)$, we have $P_{Y_1}(A) = Q_{Y_2}(\phi(A)_1)$ and $P_{Y_1}(B) = Q_{Y_2}(\phi(B)_1)$. Therefore,

$$\sum_{(y_1, y_2) \in A_1 \times B_1} P_{Y_1}^-(y_1, y_2) = \sum_{(y_1, y_2) \in \phi(A_1) \times \phi(B_1)} Q_{Y_2}^-(y_1, y_2).$$

Thus $A_1 \times B_1$ and $\phi(A_1) \times \phi(B_1)$ satisfy condition (1) in Def. IV.2.

To prove condition (2), choose $y_1 \in A_1, y_2 \in B_1$, and let $y_3 \in \phi(A_1)$ and $y_4 \in \phi(B_1)$. By Def. IV.2, there exist x_1 and x_2 such that $P_{X|Y_1}(x + x_1|y_1) = Q_{X|Y_2}(x|y_3)$ and $P_{X|Y_1}(x + x_2|y_2) = Q_{X|Y_2}(x|y_4)$ for all $x \in \mathcal{X}$. Thus

$$\begin{aligned} Q_{X|Y_2}^-(x|(y_3, y_4)) &= \sum_{u_2 \in \mathcal{X}} Q_{X|Y_2}(x + u_2|y_3) Q_{X|Y_2}(u_2|y_4) \\ &= \sum_{u_2 \in \mathcal{X}} P_{X|Y_1}(x + u_2 + x_1|y_1) P_{X|Y_1}(u_2 + x_2|y_2) \\ &= \sum_{u_2 \in \mathcal{X}} P_{X|Y_1}((x + z) + u_2|y_1) P_{X|Y_1}(u_2|y_2) \\ &= P_{X|Y_1}^-(x + z|(y_1, y_2)), \end{aligned}$$

where $z = x_1 + (-x_2)$. Therefore, $A_1 \times B_1 \simeq \phi(A_1) \times \phi(B_1)$, and $P_{XY_1} \equiv Q_{XY_2}$.

(b). (The '+' case) The distribution $P_{XY_1}^+$ and $Q_{XY_2}^+$ are over $\mathcal{X} \times (\mathcal{X} \times \mathcal{Y}_1^2)$ and $\mathcal{X} \times (\mathcal{X} \times \mathcal{Y}_2^2)$ respectively. Similarly to case (a) above, we will show that for every $A_1, B_1 \in \mathcal{Y}_1$ there exist permutations π_{y_1, y_2} and π_{y_3, y_4} on \mathcal{X} such that for every $u \in \mathcal{X}$

$$\{(\pi_{y_1, y_2}(u), y_1, y_2) : y_1 \in A_1, y_2 \in B_1\} \simeq \{(\pi_{y_3, y_4}(u), y_3, y_4) : y_3 \in \phi(A_1), y_4 \in \phi(B_1)\}$$

To show this, fix $A_1, B_1 \in \mathcal{Y}_1$ and choose some $z_1 \in A_1, z_2 \in B_1, y_1 \in A_1, y_2 \in B_1, y_3 \in \phi(A_1)$ and $y_4 \in \phi(B_1)$. By Def. IV.2, for every $x \in \mathcal{X}$ there exist x_1, x_2, x_3 and x_4 such that

$$\begin{aligned} P_{X|Y_1}(x + x_1|z_1) &= P_{X|Y_1}(x|y_1), & P_{X|Y_1}(x + x_2|z_2) &= P_{X|Y_1}(x|y_2) \\ P_{X|Y_1}(x + x_3|z_1) &= Q_{X|Y_2}(x|y_3), & P_{X|Y_1}(x + x_4|z_2) &= Q_{X|Y_2}(x|y_4). \end{aligned}$$

For $x \in \mathcal{X}$ define permutations $\pi_{y_1, y_2}, \pi_{y_3, y_4}$ as $\pi_{y_1, y_2}(x) = -x_1 + x + x_2$ and $\pi_{y_3, y_4}(x) = -x_3 + x + x_4$. We compute

$$\begin{aligned} P_{X|Y_1}^+(x|(\pi_{y_1, y_2}(u), y_1, y_2)) &= P_{X|Y_1}^+(x|(-x_1 + x_2 + u, y_1, y_2)) \\ &= \frac{P_{X|Y_1}(-x_1 + x_2 + x + u|y_1) P_{X|Y_1}(x|y_2)}{\sum_{x_0 \in \mathcal{X}} P_{X|Y_1}(-x_1 + x_2 + x_0 + u|y_1) P_{X|Y_1}(x_0|y_2)} \\ &= \frac{P_{X|Y_1}(x + x_2 + u|z_1) P_{X|Y_1}(x + x_2|z_2)}{\sum_{x_0 \in \mathcal{X}} P_{X|Y_1}(x_0 + x_2 + u|y_1) P_{X|Y_1}(x_0 + x_2|y_2)} \\ &= P_{X|Y_1}^+(x + x_2|(u, z_1, z_2)). \end{aligned}$$

Similarly,

$$Q_{X|Y_2}^+(x|(\pi_{y_3, y_4}(u), y_3, y_4)) = P_{X|Y_1}^+(x + x_4|(u, z_1, z_2)).$$

The last two equations imply that

$$P_{X|Y_1}^+(-x_2 + x_4 + x|(\pi_{y_1, y_2}(u), y_1, y_2)) = Q_{X|Y_2}^+(x|(\pi_{y_3, y_4}(u), y_3, y_4)),$$

which verifies condition (2) in Def. IV.2. Let us check that condition (1) is satisfied as well. We have

$$\begin{aligned} P_{Y_1}^+(\{(\pi_{y_1, y_2}(u), y_1, y_2) : y_1 \in A_1, y_2 \in B_1\}) &= \sum_{y_1 \in A_1, y_2 \in B_1} P_{Y_1}^+(\pi_{y_1, y_2}(u), y_1, y_2) \\ &= \sum_{y_1 \in A_1, y_2 \in B_1} P_{Y_1}(y_1) P_{Y_1}(y_2) \sum_{x \in \mathcal{X}} P_{X|Y_1}(-x_1 + x_2 + x + u|y_1) P_{X|Y_1}(x|y_2) \\ &= \sum_{y_1 \in A_1, y_2 \in B_1} P_{Y_1}(y_1) P_{Y_1}(y_2) \sum_{x \in \mathcal{X}} P_{X|Y_1}(u + x_2 + x|z_1) P_{X|Y_1}(x + x_2|z_2) \end{aligned}$$

$$= \left(\sum_{x \in \mathcal{X}} P_{X|Y_1}(u+x|z_1) P_{X|Y_1}(x|z_2) \right) \left(\sum_{y_1 \in A_1} P_{Y_1}(y_1) \right) \left(\sum_{y_2 \in B_1} P_{Y_1}(y_2) \right)$$

and

$$Q_{Y_2}^+ (\{(\pi_{y_3, y_4}(u), y_3, y_4) : y_3 \in \phi(A_1), y_4 \in \phi(B_1)\}) \\ = \left(\sum_{x \in \mathcal{X}} P_{X|Y_1}(u+x|z_1) P_{X|Y_1}(x|z_2) \right) \left(\sum_{y_3 \in \phi(A_1)} Q_{Y_2}(y_3) \right) \left(\sum_{y_4 \in \phi(B_1)} Q_{Y_2}(y_4) \right).$$

By assumption $P_{Y_1}(A_1) = Q_{Y_2}(\phi(A_1))$ and $P_{Y_1}(B_1) = Q_{Y_2}(\phi(B_1))$, so this proves that

$$P_{Y_1}^+ (\{(\pi_{y_1, y_2}(u), y_1, y_2) : y_1 \in A_1, y_2 \in B_1\}) = Q_{Y_2}^+ (\{(\pi_{y_3, y_4}(u), y_3, y_4) : y_3 \in \phi(A_1), y_4 \in \phi(B_1)\}).$$

Thus for every $u \in \mathcal{X}$

$$\{(\pi_{y_1, y_2}(u), y_1, y_2) : y_1 \in A_1, y_2 \in B_1\} \simeq \{(\pi_{y_3, y_4}(u), y_3, y_4) : y_3 \in \phi(A_1), y_4 \in \phi(B_1)\}$$

The proof is complete.

REFERENCES

- [1] E. Arkan, *Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels*, IEEE Trans. Inform. Theory **55** (2009), no. 7, 3051–3073.
- [2] E. Abbe, and E. Telatar, *Polar codes for the m-user multiple access channel*, IEEE Trans. Inform. Theory **58** (2012), no. 8, 5437–5448.
- [3] H. Mahdaviar and A. Vardy, *Achieving the secrecy capacity of wiretap channels using polar codes*, IEEE Trans. Inform. Theory, **57** (2011), no. 10, 6428–6443.
- [4] E. Sasoglu, E. Telatar, E. M. Yeh, *Polar codes for the two-user multiple-access channel*, IEEE Trans. Inform. Theory **59** (2013), no. 10, 6583–6592.
- [5] E. Arkan, *Source polarization*, Proc. IEEE Int. Symposium on Information Theory, Austin, TX, June 2010, 899–903.
- [6] S. B. Korada and R. Urbanke, *Polar codes are optimal for lossy source coding*, IEEE Trans. Inform. Theory, **56** (2010), no. 4, 1751–1768.
- [7] R. Mori and T. Tanaka, *Source and channel polarization over finite fields and Reed-Solomon matrices*, IEEE Trans. Inform. Theory, **60**(2014), no. 5, 2720–2736.
- [8] A. G. Sahebi and S. S. Pradhan, *Multilevel channel polarization for arbitrary discrete memoryless channels*, IEEE Trans. Inform. Theory, **59**(2013), no. 12, 7839–7857.
- [9] W. Park and A. Barg, *Polar codes for q-ary channels, q = 2^r*, IEEE Trans. Inform. Theory, **59**(2013), no. 2, 955–969.
- [10] S.B. Korada, E. Sasoglu, R. Urbanke, *Polar codes: Characterization of exponent, bounds, and constructions*, IEEE Trans. Inform. Theory, **56**(2010), no. 12, 6253–6264.
- [11] S.H. Hassani and R. Urbanke, *Universal polar codes*, arXiv:1307.7223, 2013.
- [12] R. Mori and T. Tanaka, *Performance and construction of polar codes on symmetric binary-input memoryless channels*, Proc. IEEE Int. Sympos. Inform. Theory, Seoul, Korea, 2009, 1496–1500.
- [13] I. Tal and A. Vardy, *How to construct polar codes*, IEEE Trans. Inform. Theory, **59**(2013), no. 10, 6562–6582.
- [14] R. Pedarsani, S.H Hassani, I. Tal, E. Telatar, *On the construction of polar codes*, Proc. IEEE Int. Sympos. Inform. Theory, St. Petersburg, Russia, 2011, 11–15.
- [15] E. Sasoglu, *Polarization and Polar Codes*, Foundations and Trends in Communications and Information Theory. vol. 8, Now Publishers, 2012.
- [16] A. Ghayoori and T.A. Gulliver, *Constructing polar codes using iterative bit-channel upgrading*, arXiv:1302.5153, 2013.
- [17] I. Tal, A. Sharov, A. Vardy, *Constructing polar codes for non-binary alphabets and MACs*, Proc. IEEE Int. Sympos. Inform. Theory, Boston, MA, 2012, 2132–2136.
- [18] U. Pereg and I. Tal, *Channel upgradation for non-binary input alphabets and MACs*, Proc. IEEE Int. Sympos. Inform. Theory, Honolulu, HI, 2014, 411–415.
- [19] A. Ghayoori and T.A. Gulliver, *Upgraded approximation of non-binary alphabets for polar code construction*, arXiv:1304.1790, 2013.
- [20] P. Trifonov, *Efficient design and decoding of polar codes*, IEEE Trans. on Comm., **60**(2012), no. 11, 3221–3227.
- [21] H. Li and J. Yuan, *A practical construction method for polar codes in AWGN channels*, in TENCON Spring Conference, Sydney, NSW, 2013, 223–226.
- [22] D. Wu, Y. Li, Y. Sun, *Construction and block error rate analysis of polar codes over AWGN channel based on Gaussian approximation*, IEEE Comm. Letters, **18**(2014), no. 7, 1099–1102.
- [23] H. Vangala, E. Viterbo, Y. Hong, *A comparative study of polar code constructions for the AWGN channel*, arXiv:1501.02473, 2015.
- [24] D. Kern, S. Vorkoper, V. Kuhn, *A new code construction for polar codes using min-sum density*, in ISTC, 8th, Bremen, Germany, 2014, 228–232.
- [25] G. Bonik, S. Goreinov, N. Zamarashkin, *Construction and analysis of polar and concatenated polar codes: practical approach*, arXiv:1207.4343, 2012.
- [26] S. Zhao, P. Shi, B. Wang, *Designs of Bhattacharyya parameter in the construction of polar codes*, in WiCOM, 7th, Wuhan, China, 2011, 1–4.
- [27] A. Bravo-Santos, *Polar codes for the Rayleigh fading channel*, IEEE Comm. Letters, **17**(2013), no. 12, 2352–2355.
- [28] K. Chen, K. Niu, J.R. Lin, *Practical polar code construction over parallel channels*, IET Comm., **7**(2013), no. 7, 620–627.
- [29] L. Zhang, Z. Zhang, X. Wang, *Polar code with blocklength N = 3ⁿ*, in WCSP, Huangshan, China, 2012, 1–6.
- [30] V. Miloslavskaya and P. Trifonov, *Design of binary polar codes with arbitrary kernel*, in ITW, Lausanne, Switzerland, 2012, 119–123.
- [31] B. Serbetci and A.E. Pusane, *Practical polar code construction using generalised generator matrices*, IET Comm., **8**(2014), no. 4, 419–426.
- [32] P. Trifonov and P. Semenov, *Generalized concatenated codes based on polar codes*, in ISWCS, 2011, 442–446.
- [33] H. Mahdaviar, M. El-Khamy, J. Lee, I. Kang, *On the construction and decoding of concatenated polar codes*, Proc. IEEE Int. Sympos. Inform. Theory, Istanbul, Turkey, 2013, 952–956.
- [34] I. Tal, *On the construction of polar codes for channels with moderate input alphabet sizes*, arXiv:1506.08370, 2015.
- [35] E. Arkan and E. Telatar, *On the rate of channel polarization*, Proc. IEEE Int. Sympos. Inform. Theory, Seoul, Korea, 2009, 1493–1495.

- [36] E. Sasoglu, E. Telatar, E. Arikan, *Polarization for arbitrary discrete memoryless channels*, in ITW, Taormina, Italy, 2009, 144–148.
- [37] R. Nasser, *Ergodic theory meets polarization. I: An ergodic theory for binary operations*, arXiv:1406.2943, 2014.
- [38] R. Nasser, *Ergodic theory meets polarization. II: A foundation of polarization theory*, arXiv: 1406.2949, 2014.
- [39] E. Sasoglu, *Polar codes for discrete alphabets*, Proc. IEEE Int. Sympos. Inform. Theory, Boston, MA, 2012, 2137–2141.
- [40] T. Cover and J. Thomas, *Elements of Information Theory*, Wiley-Interscience, 1991.